课程设计

第一部分：相控阵雷达仿真

一、 仿真内容

（1） 模拟 LFM 脉冲信号（带宽 1MHz，脉宽 100us），画出其时频域波形和模糊函数；

（2） 模拟目标回波（目标距离 90km，速度 60m/s），进行匹配滤波仿真（无噪声）；

（3） 设雷达的脉冲重复频率为 1KHz，工作载频 1GHz，模拟脉冲串回波信号（含噪声，信噪比-10dB），并进行匹配滤波和 MTD 处理，给出 MTD 输出的“距离-速度-幅度”三维图，标出目标的距离-速度信息；

（4） 画出阵列天线（16 阵元均匀线阵）方向图，采用 hamming 窗控制旁瓣；模拟阵列接收信号，对其进行 DBF 处理仿真，给出“DBF-脉冲压缩-MTD”处理输出的“距离-速度-幅度”三维图，标出目标。

二、 信号模型与算法公式

1. 线性调频（LFM）信号

脉冲压缩雷达能同时提高雷达的作用距离和距离分辨率。这种体制采用宽脉冲发射以提高发射的平均功率，保证足够大的作用距离；而接收时采用相应的脉冲压缩算法获得窄脉冲，以提高距离分辨率，较好的解决雷达作用距离与距离分辨率之间的矛盾。

脉冲压缩雷达最常见的调制信号是线性调频（Linear Frequency Modulation，LFM）信号，接收时采用匹配滤波器（Matched Filter）压缩脉冲。

LFM 信号（也称 Chirp 信号）数学表达式为：

$$s(t)=rect\left(\frac{t}{T}\right)e^{j2\pi\left(f_c t+\frac{K}{2}t^2\right)} \tag{1.1}$$

其中 f_c 为载波频率，$rect\left(\frac{t}{T}\right)$ 为矩形信号，$K=\frac{B}{T}$ 为调频斜率，由此信号瞬时频率为 $\left[f_c-K\frac{T}{2},f_c+K\frac{T}{2}\right]$。可重写为

$$s(t)=S(t)e^{j2\pi f_c t} \tag{1.2}$$

其中

$$S(t)=rect\left(\frac{t}{T}\right)e^{j\pi Kt^2} \tag{1.3}$$

为 LFM 信号复包络，具有与时域信号相同的幅频特性。因此，简洁起见，仅需用式(1.3)就能构建 LFM 时域信号。MATLAB 里可绘制 LFM 信号的时频特性曲线，调用 ambgfun()函数即可

绘制 LFM 信号的模糊函数。

2. 点目标回波信号

假设点目标距离雷达收发机 $R_t = 90\text{km}$，点目标速度 $v_t = 60\text{m/s}$，雷达发送 LFM 信号，构建脉冲间隔内时间序列 $[0,T]$，设定信号采样率 $f_s = 5B$，可以计算点目标时延

$$t_d = \frac{2R_t}{c} \tag{1.4}$$

及多普勒频移

$$f_d = \frac{2v_t f_c}{c} \tag{1.5}$$

由于 LFM 信号特性，回波信号主要由时延与相移构成，为

$$s_r(t) = s(t) * h(t) = \sigma_t s(t - t_d) \tag{1.6}$$

简洁起见，一般反射特性设置 $\sigma_t = 1$，仅考虑时延与相移。

通过匹配滤波，输出最大信噪比时刻及对应目标时延信号 $y(t)$

$$y(t) = T\frac{\sin \pi KT\left(1 - \frac{|t|}{T}\right)t}{\pi KTt} rect\left(\frac{t}{2T}\right)e^{j2\pi f_c t} \tag{1.7}$$

3. 脉冲串回波信号

设雷达发射脉冲串信号的 PRF 为1kHz，载频 $f_c = 1\text{GHz}$，则多普勒频移为

$$f_D = \frac{2v_t}{\lambda} = \frac{2v_t f_c}{c} \tag{1.8}$$

每个 PRT 内发送单脉冲 i，其回波的多普勒相位可以表示为

$$\theta_{D_i} = j2\pi f_D\left(\frac{i}{PRF} + t_d\right) \tag{1.9}$$

设定累积脉冲数为 N_{pulse}，可得每个 PRT 内脉冲 i 的回波信号可表示为

$$Echo_i = s(t) \cdot e^{\theta_{D_i}} \tag{1.10}$$

对所有脉冲叠加得到

$$\boldsymbol{Echo} = \left[Echo_1, Echo_2, \ldots, Echo_{N_{pulse}}\right]^T \tag{1.11}$$

再叠加加性复高斯白噪声，其信噪比为SNR，得到最终脉冲串接收信号

$$\boldsymbol{Received} = \boldsymbol{Echo} + \boldsymbol{AWGN} = \boldsymbol{Echo} + \sqrt{\frac{P_{rx}}{2 \cdot SNR}} e^{j\boldsymbol{rand}(size(\boldsymbol{Echo}))} \tag{1.12}$$

信号大小为$\mathbb{C}^{(N_{pulse}\times N_{PRT})\times 1}$，其中$P_{rx}$为***Echo***的平均功率，$N_{PRT}=PRT\cdot f_s$为PRT内的采样点数。

MTD处理过程即为先后对距离与多普勒进行FFT，即可得到距离-多普勒幅度谱$\Upsilon(\ell,F_D)$。

绘制距离-速度-幅度三维图时，根据距离分辨率

$$\Delta R=\frac{c}{2}\cdot\frac{N_{PRT}}{B} \tag{1.13}$$

以及速度分辨率

$$\Delta v=\frac{\lambda}{2}\cdot\frac{PRF}{N_{pluse}} \tag{1.14}$$

可分别计算得到距离轴和速度轴范围。特别地，对接收信号进行匹配滤波后，应注意截断无效区域（对应距离轴负距离）。

4. 均匀线阵阵列接收信号

对比单天线脉冲串接收信号，均匀线阵阵列接收信号建模的区别在于在每个脉冲内依次对每个阵元计算相位补偿和时延，其中时延t_d与单天线相同。对于每个脉冲i里的每个阵元k，其相位补偿修改为

$$Phase_k=Array\ Phase_k\bullet e^{\theta_{D_i}} \tag{1.15}$$

其中

$$Array\ Phase_k=e^{-j2\pi\frac{d}{\lambda}\sin\delta_t(k-1)} \tag{1.16}$$

δ_t为目标方位角。最终得到均匀线阵阵列接收信号，大小为$\mathbb{C}^{(N_{pulse}\times N_{PRT})\times N_R}$。

5. DBF波束形成

DBF采用加汉明窗进行波束形成，通过抑制旁瓣幅度来增强主瓣显示。MATLAB里可通过 hamming()函数调用汉明窗加权向量。设权重向量$\boldsymbol{w}=\left[w_0,w_1,\ldots,w_{N_R-1}\right]^T$，其中参数设定阵元数$N_R=16$，以在阵元维度进行波束形成，即

$$\boldsymbol{y}(t)=\boldsymbol{w}^H\boldsymbol{x}(t)=\boldsymbol{w}^H\boldsymbol{a}_s(\theta)s(t) \tag{1.17}$$

其中$\boldsymbol{a}_s(\theta)=e^{j\phi\boldsymbol{N}_R}=e^{j\phi[0,1,\ldots,N_R-1]^T}$为空域导向矢量，$\phi=\dfrac{2\pi d\sin\delta_t}{\lambda}$为归一化空间角频率。若$\boldsymbol{w}^H\boldsymbol{a}_s(\theta)=0$，则$\boldsymbol{y}(t)=0$，若$\boldsymbol{w}=\boldsymbol{a}_s(\theta)$，则$\boldsymbol{y}(t)=As(t)$。依据此可以对阵列接收信号进行DBF处理，构造汉明窗加权的归一化权向量

$$\boldsymbol{w}=\frac{\boldsymbol{Array\ Phase}\cdot\boldsymbol{w}_{hamming}}{\boldsymbol{Array\ Phase}^H\cdot\boldsymbol{Array\ Phase}\cdot\boldsymbol{w}_{hamming}} \tag{1.18}$$

来保持主瓣增益，还可使得阵列天线方向图在强干扰方向处形成凹口，即干扰置 0。输出包含累积脉冲数和 PRT 采样数两个维度的波形数据，再按照单天线脉冲串回波信号处理的流程即可得到幅度谱 $\Upsilon\left(\ell,F_D\right)$。虽然汉明窗加权后可降低副瓣电平至 -40dB 以下，但是其也会导致主瓣展宽。

三、 仿真结果与结果分析

依次按照仿真实验内容，在 MATLAB R2023b 平台进行仿真实验，所有仿真参数已在附录 A 表格给出。绘制 LFM 时域与频率图（见图 1.1、图 1.2），以及模糊函数三维图和等高线图（图 1.3、图 1.4）。

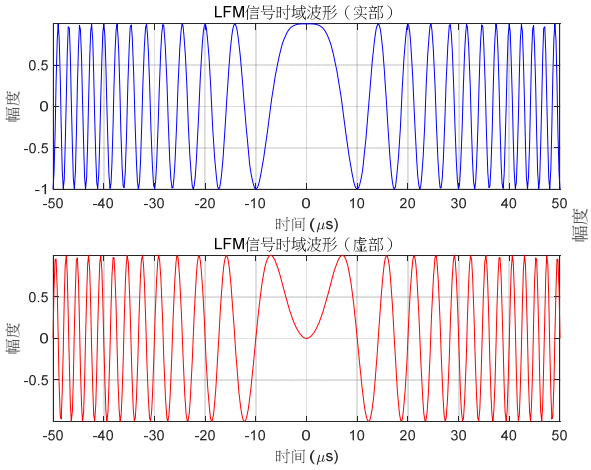


图 1.1 LFM 时域波形

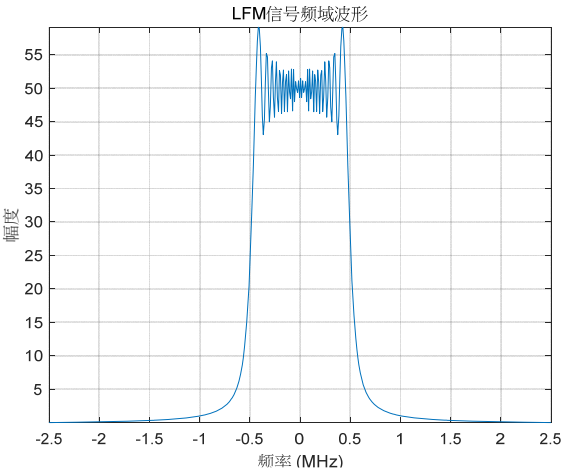


图 1.2 LFM 频域波形

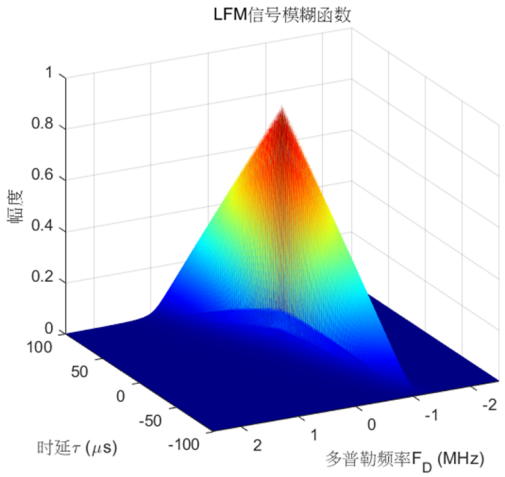


图 1.3 LFM 模糊函数

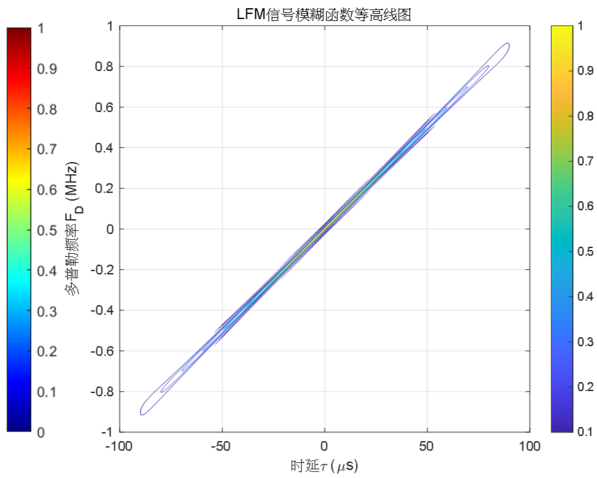


图 1.4 LFM 模糊函数等高线

可以计算得到 LFM 时间带宽积为 $B\tau=100$，在 5 倍带宽的采样率下，LFM 频谱在 $\pm B/2$ 范围内具有趋于矩形的现象，这与 PSP（驻相原理）近似频谱在该范围内具有恒定值、而在该范围外具有零值是一致的。LFM 时域表现类似正弦波声音信号，且随着时间的增加频率也随之增加。这与理论是完全符合的。

观察 LFM 模糊图可以发现，其分布形如立起来的三角形，且通过等高线图可以发现，其

在时延-多普勒平面上有一定角度的倾斜，形状近似很窄的椭圆，旁瓣很低。随着多普勒失配的增加，时延也随之增加，但峰值却越来越低。这反映了大时间带宽积的 LFM 信号解决了距离分辨率与速度分辨率的矛盾，且均具有较高的分辨率。

在点目标的回波模拟（无噪声）仿真中，图 2.1 反映了点目标回波信号的时域波形是延迟了 0.6ms 后的 LFM 信号，实际延迟时间与理论计算 $t_d = 2\times 90\times 10^3 / 3\times 10^8\,\mathrm{s} = 0.6\mathrm{ms}$ 相符。通过匹配滤波器后，观察图 2.2 发现该回波信号在 90km 处出现尖峰，且旁瓣比很高。说明回波信号经过匹配滤波后能够在距离上分辨出目标所在位置。根据距离分辨率公式计算 $\Delta R = 3\times 10^8/2 = 150\times 10^6\,\mathrm{m}$，对应目标位置 $R_t = t_d \bullet \Delta R = 0.6\times 10^{-3}\times 150\times 10^6\,\mathrm{m} = 90\mathrm{km}$，仿真验证了理论计算的正确性。进一步观察发现，主瓣宽度被压缩成窄脉冲，能量汇集在 90km 处，且由于本身时间带宽积大，目标探测的分辨率很高，探测性能很强。输出窄脉冲的宽度与理论计算 $c/2B = 150\mathrm{m}$ 相同。

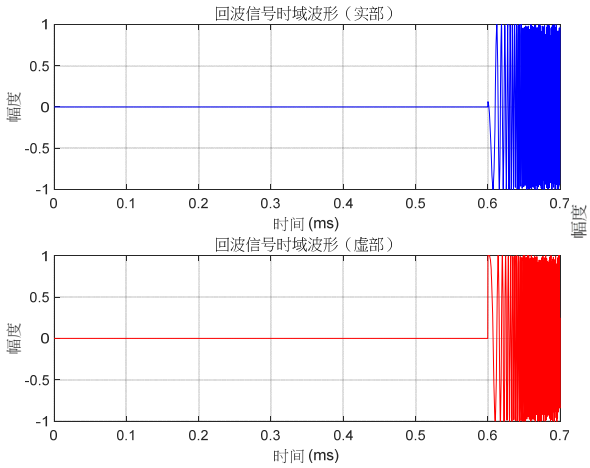


图 2.1 点目标回波时域波形

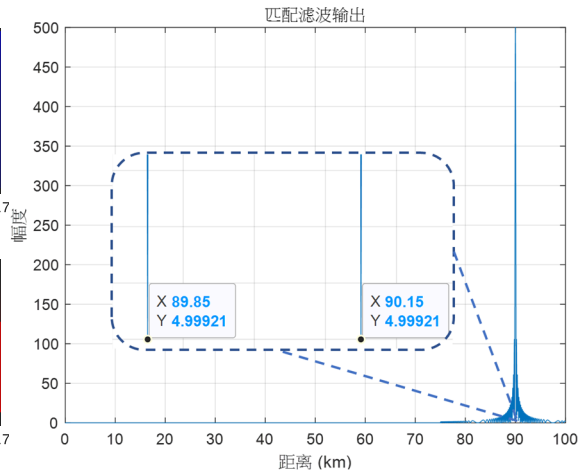
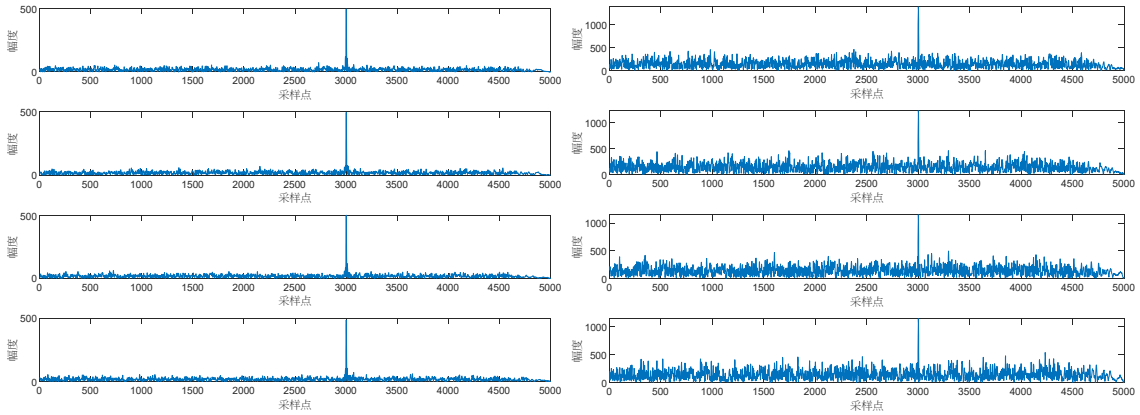


图 2.2 点目标回波信号匹配滤波输出

在脉冲串回波信号仿真中，通过调用脉冲串接收函数得到无噪声的回波函数，叠加加性复高斯白噪声后得到含噪声的脉冲串回波接收信号。



(a) 匹配滤波后波形　　(b) MTD 后波形

图 3.1 脉冲串回波信号各环节处理后波形

先后进行匹配滤波（图 3.1（a））和 MTD 处理（图 3.1（b）），即先后从距离维度和多普勒频移维度突显目标距离和速度信息，观察发现主瓣与旁瓣一直保持着较高的峰值旁瓣比（PSLR），且最低 PSLR 约 9.17dB 。

MTD 目标显示如图 3.2 所示。可以发现目标距离检测一致，而速度检测的相对误差为 1.5% 。

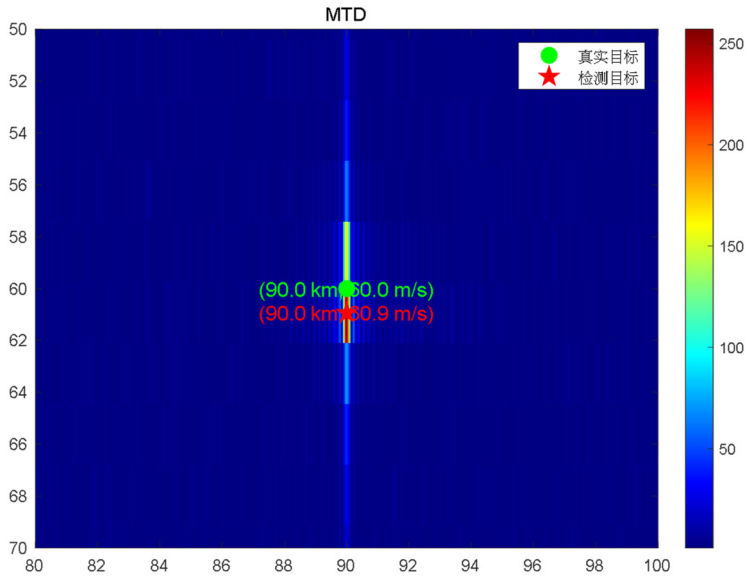


图 3.2 MTD 目标显示

绘制 MTD 输出的距离-速度-幅度谱三维图，如图 3.3 所示。检测目标所在距离位置出现明显窄波，表明距离分辨率很高，能够精确定位目标位置；而该窄波在速度为 60.9m/s 处出现尖峰。噪声被抑制在 -40dB 以下。表明匹配滤波与 MTD 处理的有效性。尖峰在速度轴不出现在 60m/s 处可能是多普勒频率 FFT 产生量化误差导致的。

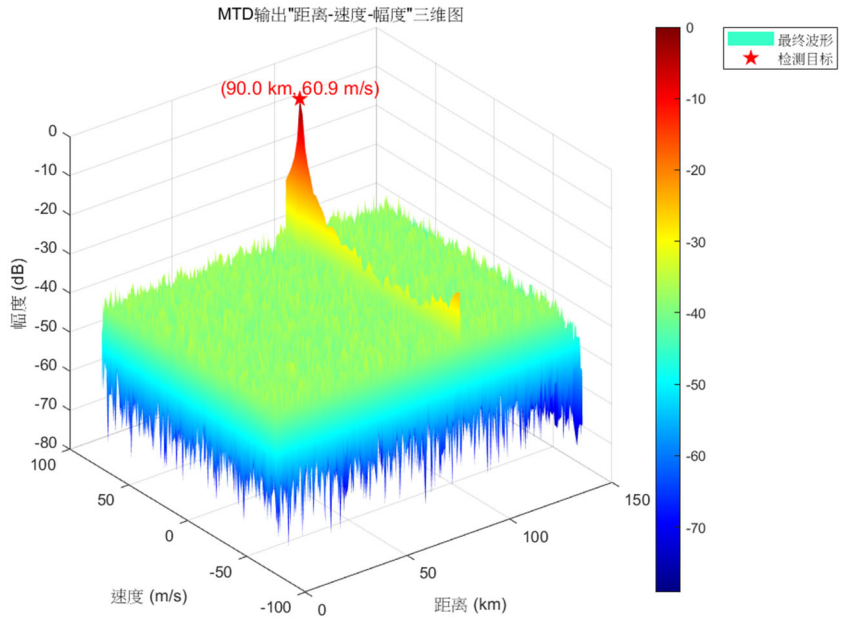


图 3.3 MTD 输出距离-速度-幅度三维图

现在尝试计算量化误差范围。仿真中累积脉冲数 N_{pluse} 为 64，PRF 为 1kHz ，多普勒频率

通过 FFT 进行估计，根据 FFT 频率分辨率 $\Delta f = PRF/N_{pluse}$ 以及速度分辨率 $\Delta v = \frac{\lambda}{2}\Delta f$ 可以计算得到 $\Delta v \approx 2.34\text{m/s}$。由于目标真实速度对应于多普勒频率并未对齐 FFT 频点，最大幅度点可能会落在最近邻频点而导致检测量化误差，所以可以得到速度轴的量化误差范围即为 $\pm 1.17\text{m/s}$，其对应相对误差 $\pm 1.95\%$。仿真相对误差 1.5% 恰好在该范围内，验证了这一猜想。

在相控阵雷达 16 阵元均匀线阵仿真中，通过汉明窗对阵列天线方向图进行加窗，如图 4.1 所示。可以发现加窗后 PSLR 从 13.15dB 增加到 39.37dB，然而引发了主瓣展宽，从 7.2° 增加到 17.4°，也损失了 SNR。

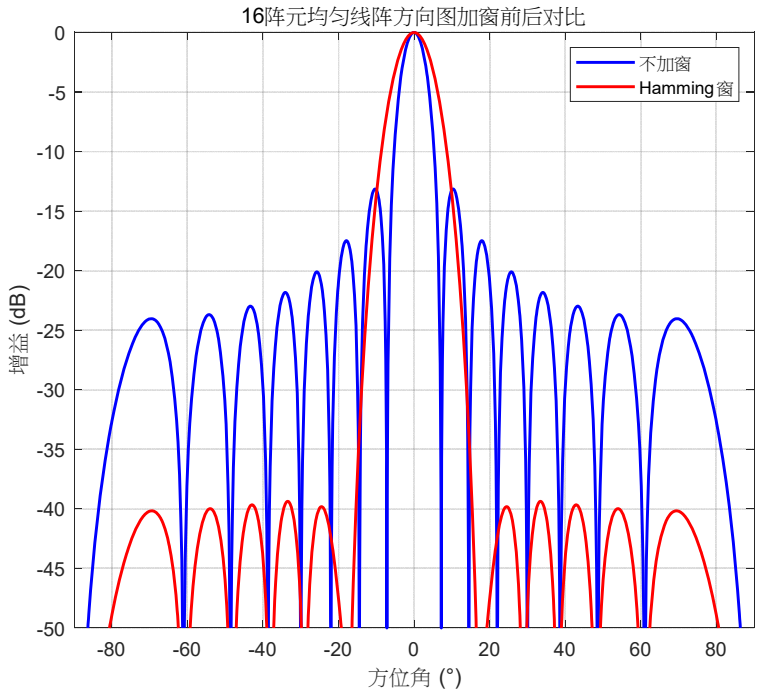


图 4.1 16 阵元均匀线阵阵列天线方向图

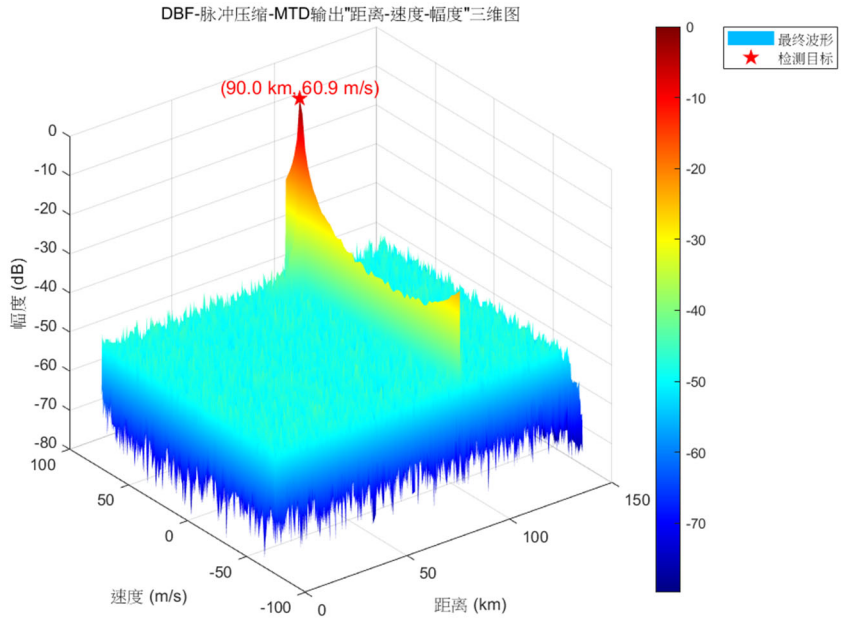


图 4.2 DBF-脉冲压缩-MTD 输出距离-速度-幅度三维图

进行 DBF 处理后，与前面仿真处理过程一致，绘制输出三维图（图 4.2），噪声干扰被明

显抑制在 -50dB 以下，对比图 3.3 的干扰抑制效果更加明显。虽然经过加窗后相控阵雷达检测目标性能更好，但是会导致波形产生畸变，改变信号频谱特征，甚至可能发生能量泄漏现象。但是，在杂波等干扰背景下，由于高副瓣将影响雷达对目标的检测，容易产生虚警，因此往往都会采用加窗技术进行旁瓣抑制，只是需要根据需要谨慎选择合适的窗函数，在下一个部分仿真实验将会体现。

四、 附录 A

第一部分用到的仿真参数如下表所示。

参数	符号	数值	单位
光速	c	3×10^8	m/s
带宽	B	1	MHz
脉宽	T_{au}	100	μs
采样率	f_s	5	MHz
目标距离	R_t	90	m
目标速度	v_t	60	m/s
LFM 持续时间	t	100	μs
载波频率	f_c	1	GHz
脉冲重复频率	PRF	1	kHz
脉冲重复间隔	PRT	1	ms
波长	λ	0.3	m
信噪比	SNR	−10	dB
多普勒频移	f_D	400	Hz
脉冲数	N_{pluse}	64	-
阵元数	N_R	16	-
阵元间距	d	0.15	m
目标方位角	δ_t	0	°

仿真实验中用到的主函数为 test1.m，自定义函数有：received_pulseburst.m 和 received_array.m。

received_pulseburst.m

```
%% 生成脉冲串接收信号
function received = received_pulseburst(St, num_pulses, fd, PRF, td, N_PRT, N_d, N_st)
    % 参数设置
    % St LFM 发送信号
    % num_pulses 脉冲数
    % fd 多普勒频移
```

```
    % PRF 脉冲重复频率
    % td 目标延迟时间
    % N_PRT 单个 PRT 的采样点数
    % N_d 目标时延的采样点数
    % N_st 单个脉冲内的采样点数
    % 返回脉冲串接收信号 （脉冲数×单个 PRT 的采样点数，1）

    % 生成脉冲串接收信号
    received = zeros(num_pulses * N_PRT, 1);
    for n = 0:num_pulses-1
        % 生成单脉冲回波（含多普勒相位和时间延迟）
        doppler_phase = exp(1j * 2 * pi * fd * (n / PRF + td));
        % 计算回波在接收窗口中的位置
        start_idx = n * N_PRT + N_d;
        end_idx = start_idx + N_st;

        % 截断处理防止越界
        if end_idx > num_pulses * N_PRT
            end_idx = num_pulses * N_PRT;
            valid_len = end_idx - start_idx;
            received(start_idx+1:end_idx) = received(start_idx+1:end_idx) + ..
.
                St(1:valid_len).' * doppler_phase;
        else
            received(start_idx+1:end_idx) = received(start_idx+1:end_idx) + ..
.
                St.' * doppler_phase;
        end
    end
end
```

received_array.m

```
%% 生成阵列接收信号
function rx_array_signal = received_array(St, array_phase, N_R, num_pulses, fd
, PRF, td, N_PRT, N_d, N_st)
    % 参数设置
    % St LFM 发送信号
    % array_phase 阵列相位
    % N_R 阵元数
    % num_pulses 脉冲数
    % fd 多普勒频移
    % PRF 脉冲重复频率
    % td 目标延迟时间
    % N_PRT 单个 PRT 的采样点数
```

```
    % N_d 目标时延的采样点数
    % N_st 单个脉冲内的采样点数
    % 返回阵列接收信号 （脉冲数×单个 PRT 的采样点数，阵元数）

    % 初始化多通道接收信号
    rx_array_signal = zeros(num_pulses * N_PRT, N_R);
    for n = 0:num_pulses-1
        % 生成单脉冲回波（含多普勒相位和时间延迟）
        doppler_phase = exp(1j * 2 * pi * fd * (n / PRF + td));
        % 生成阵列接收信号
        for k = 1:N_R
            % 每个阵元的相位补偿
            R_phase = array_phase(k) * doppler_phase;
            % 计算信号位置
            start_idx = n * N_PRT + N_d;
            end_idx = start_idx + N_st;
            % 截断处理
            if end_idx > num_pulses * N_PRT
                end_idx = num_pulses * N_PRT;
                valid_len = end_idx - start_idx;
                rx_array_signal(start_idx+1:start_idx+valid_len, k) = ...
                    rx_array_signal(start_idx+1:start_idx+valid_len, k) + ...

                    St(1:valid_len).' * R_phase;
            else
                rx_array_signal(start_idx+1:end_idx, k) = ...
                    rx_array_signal(start_idx+1:end_idx, k) + ...
                    St.' * R_phase;
            end
        end
    end
end
```

课程设计

第二部分：机载雷达仿真

一、 仿真内容

机载雷达采用 16 阵元均匀线阵，单阵元功率为 2kw；载机高度 5km，载机速度 150m/s；发射线性调频信号，载频 1GHz，带宽 1MHz，脉宽 100 us，脉冲重复频率 1KHz，积累脉冲数 8~256（确保检测到目标的条件下自定）；地面目标距离 90km，RCS 为 5m^2，径向速度为 60m/s。

（1） 针对正侧视阵，采用地面散射单元累加法进行杂波建模，画出 90km 处单距离环杂波的空时谱，即空时处理的结果（“空间频率-多普勒频率-幅度”三维图）；

（2） 设标准温度为 290K，杂波后向散射系数为 0.01，模拟接收信号（含目标回波、杂波和噪声），对接收信号进行匹配滤波、波束形成和脉冲积累处理（酌情使用窗函数），绘制输出“距离-速度-幅度”三维图，标出目标点，完成 CFAR 检测，提取目标的距离-速度信息；

（3） 改变目标速度（归一化多普勒频率范围-0.5~0.5），绘制输出 SCNR 曲线。

二、 信号模型与算法公式

1. 杂波信号建模

采用地面散射单元累加法对杂波信号进行建模。假设相控阵雷达各方向天线收发增益 G_t 和 G_r 均为 1，雷达接收机的系统损耗系数为 L_s，根据雷达方程，可计算发射 LFM 后得到目标回波信号的接收功率为

$$P_r = \frac{P_t N_R^2 \lambda^2 \sigma_c}{(4\pi)^3 R_t^4 L_s} \tag{2.1}$$

可计算杂波距离环间隔为

$$\delta_R = \frac{c}{2B} \tag{2.2}$$

对于所有 k_{sam} 个距离环，第 k 个距离环的俯仰角为

$$\theta = \arcsin\left(\frac{H_c}{R_t + \delta_R k}\right) \tag{2.3}$$

其中 $k \in \left[-\frac{k_{sam}}{2}, \frac{k_{sam}}{2}\right]$。

已知方位角间隔为

$$\Delta\theta = \frac{\pi}{N_{bin} - 1} \tag{2.4}$$

那么可计算第 k 个距离环所对应的机载雷达阵列天线的地面投影距离为

$$R_{ground} = \left(R_t + \delta_R k\right)\cos\left(\theta\right) \tag{2.5}$$

已知正侧视下 N_{bin} 个杂波块方位角分布在 $\boldsymbol{\varphi}_c = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ 内，分别对每个距离环 k 计算各个杂波块的 RCS 和幅度，那么第 k 个距离环的各个杂波块的面积为

$$A_c = \delta_R \cdot R_{ground} \cdot \Delta\theta \tag{2.6}$$

由此杂波块 RCS 为

$$\sigma_c = \sigma^0 A_c \tag{2.7}$$

代入式(2.1)得到接收功率，进一步计算杂波信号幅度为

$$A_c = \sqrt{P_r} \tag{2.8}$$

利用空时导向矢量

$$\boldsymbol{a}_s = e^{j2\pi \boldsymbol{N}_R \cdot \boldsymbol{f}_s},\ \boldsymbol{f}_s = \frac{d}{\lambda}\cos\left(\theta\right)\sin\left(\boldsymbol{\varphi}_c - \varphi_{array}\right) \tag{2.9}$$

$$\boldsymbol{a}_t = e^{j2\pi \boldsymbol{L} \cdot \boldsymbol{f}_d},\ \boldsymbol{f}_d = \frac{2v_c}{\lambda}\cos\left(\theta\right)\sin\left(\boldsymbol{\varphi}_c\right) / PRF \tag{2.10}$$

其中杂波块方位角矢量 $\boldsymbol{\varphi}_c = \left[-\frac{\pi}{2}, -\frac{\pi}{2}+\frac{\pi}{N_{bin}-1}, \ldots, 0, \ldots, \frac{\pi}{2}-\frac{\pi}{N_{bin}-1}, \frac{\pi}{2}\right]$，阵元矢量 $\boldsymbol{N}_R = \left[0,1,\ldots,N_R-1\right]^T$，脉冲数矢量 $\boldsymbol{L} = \left[0,1,\ldots,\mathrm{CPI}-1\right]^T$。利用累加法可得到第 k 个杂波距离环的杂波空时信号矢量

$$\boldsymbol{x}_c = \sum_{N_{bin}}\left(A_c \cdot \boldsymbol{a}_s \otimes \boldsymbol{a}_t\right) \in \mathbb{C}^{(N_R \times CPI)\times 1} \tag{2.11}$$

迭代计算最终得到所有杂波距离环的杂波空时信号矩阵 $\boldsymbol{x}_{clutter} \in \mathbb{C}^{(N_R \times CPI)\times(k_{sam}+1)}$。

2. 机载雷达均匀线阵阵列接收信号建模与处理

对比第一部分的相控阵雷达的均匀线阵阵列接收信号，机载雷达建模时只需要增加考虑实际幅度的计算即可。根据雷达功率方程(2.1)，可以计算单个阵元的接收信号幅度为

$$A_r = \sqrt{P_r} \tag{2.12}$$

那么仅需要在每个脉冲内的每个阵元生成阵列接收信号的过程中，修正接收信号为

$$s'\left(t\right) = A_r s\left(t\right) \tag{2.13}$$

即可正确生成机载雷达均匀线阵阵列接收信号。

3. CA-CFAR 检测

二维 CA-CFAR（Cell Averaging CFAR）通过对雷达信号的距离-多普勒二维数据矩阵进行

滑动窗口检测，动态估计局部噪声基底，并根据虚警概率设置自适应阈值，实现在复杂环境中对目标的恒虚警率检测。设保护单元窗口和参考单元窗口分别为$[G_r, G_d]$和$[T_r, T_d]$，有参考单元总数

$$N_{ref} = (2T_r + 2G_r + 1)(2T_d + 2G_d + 1) - (2G_r + 1)(2G_d + 1) \tag{2.14}$$

根据虚警概率P_{fa}和参考单元总数N_{ref}，可计算阈值因子

$$\alpha = N_{ref}\left(P_{fa}^{-1/N_{ref}} - 1\right) \tag{2.15}$$

具体算法流程如下表所示

CA-CFAR 算法框架

1. 初始化检测矩阵大小为$(N_{range} \times N_{speed})$；
2. 计算阈值因子α；
3. 遍历每个待检测单元(CUT)：
   a. 定义检测窗口

$$Range\ Window:\ [i - T_r - G_r, i + T_r + G_r]$$

$$Doppler\ Window:\ [j - T_d - G_d, j + T_d + G_d]$$

   b. 提取包含保护单元和参考单元的局部窗口

$$RefCells = Data(Range\ Window, Doppler\ Window)$$

   c. 将保护单元置为NaN，去除保护单元

$$RefCells(T_r + 1 : T_r + 2G_r + 1, T_d + 1 : T_d + 2G_d + 1) = NaN$$

   d. 计算噪声基底与阈值

$$\mu = mean(RefCells)$$

$$T = \alpha\mu$$

   e. 检测判决，若 CUT 值大于阈值，则标记为检测目标；
4. 定位真实目标坐标，判断真实目标是否被检测到，若未检测到，则搜索最近邻点；
5. 计算检测目标的距离和速度相对误差。

三、 仿真结果与结果分析

依次按照仿真实验内容，在 MATLAB R2023b 平台进行仿真实验，所有仿真参数已在附录 B 表格给出。调用编写的杂波函数生成所有距离环杂波信号矩阵，选取第一个杂波距离环的杂波数据进行空时二维 FFT 处理，频点个数均为 512。绘制出如图 5.1 的三维空时谱，在此基础上计算理论杂波脊并绘制在三维图中。同时观察空时谱的空间频率-多普勒频率平面图（图 5.2），可以发现正侧视场景下，单杂波距离环实际杂波脊倾斜斜率与理论杂波脊倾斜斜率一致，且均

为 $k_{clutter}=2v_c/(d\bullet PRF)=2\times150/0.15\times10^3=2$，表明正侧视场景下，固定的单杂波距离环下不同杂波块的多普勒频率正比于方位角的正弦值（也正比于空间频率），信号幅度则受到阵列发射方向图的空时调制。

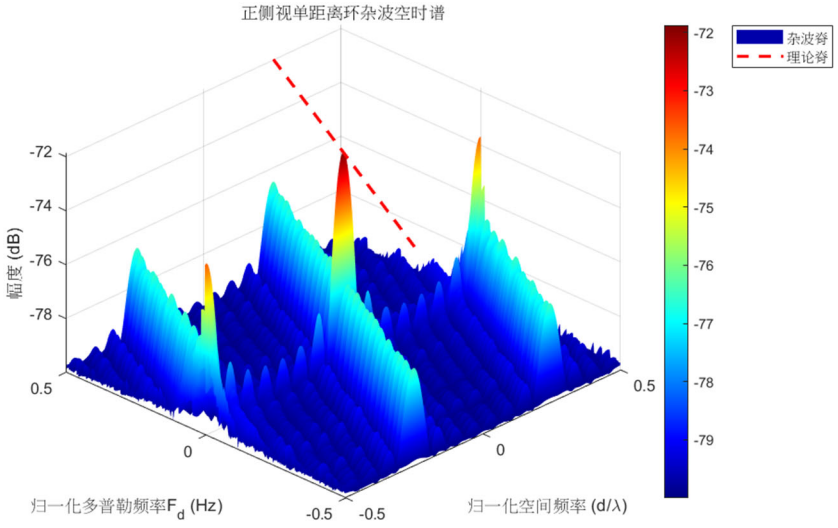


图 5.1 正侧视单距离环杂波空时谱

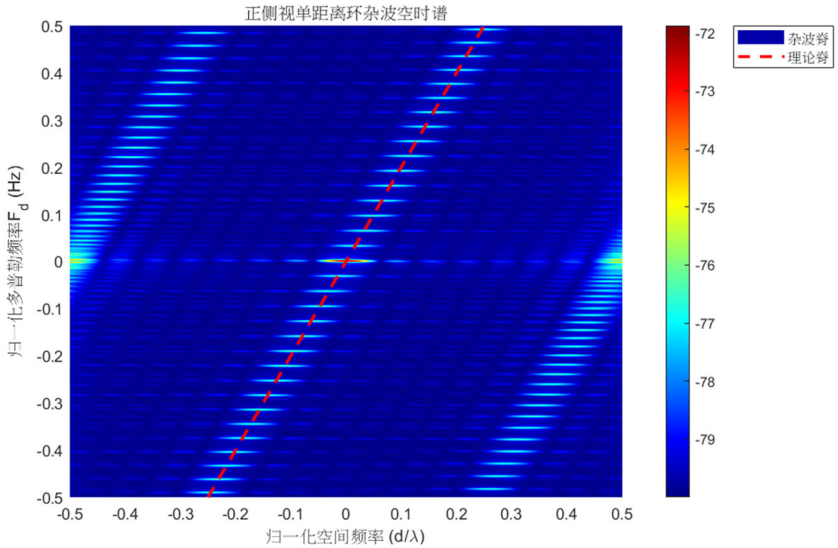


图 5.2 正侧视单距离环杂波空时谱的空间频率-多普勒频率平面图

但该杂波建模过程可能存在一定的问题，观察幅度谱发现其最大幅度值为-72dB，大部分分布在 $(-80\text{dB},-76\text{dB})$ 内，这可能是因为其中一些参数的设定不太实际。

在机载雷达均匀线阵阵列接收信号处理仿真中，调用机载雷达阵列接收信号函数得到目标回波信号，叠加杂波与加性高斯白噪声得到最终的接收信号矩阵，其形状为 $\mathbb{C}^{(N_R\times N_{pulse})\times N_R}$。类似地，按照 DBF-匹配滤波-MTD 顺序对该接收信号进行处理，绘制出距离-速度-幅度三维图，如图 6.1 和图 6.2 所示。可以发现主要由两个互相垂直的窄波组成脉冲信号，而幅度最大值所在平面坐标分别对应机载雷达接收机检测到目标的距离和速度信息。

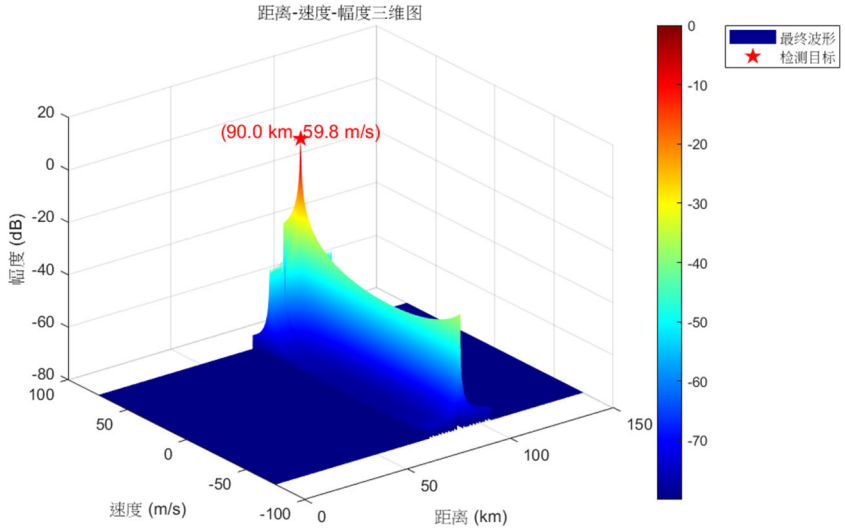


图 6.1 机载雷达接收信号处理后距离-速度-幅度谱

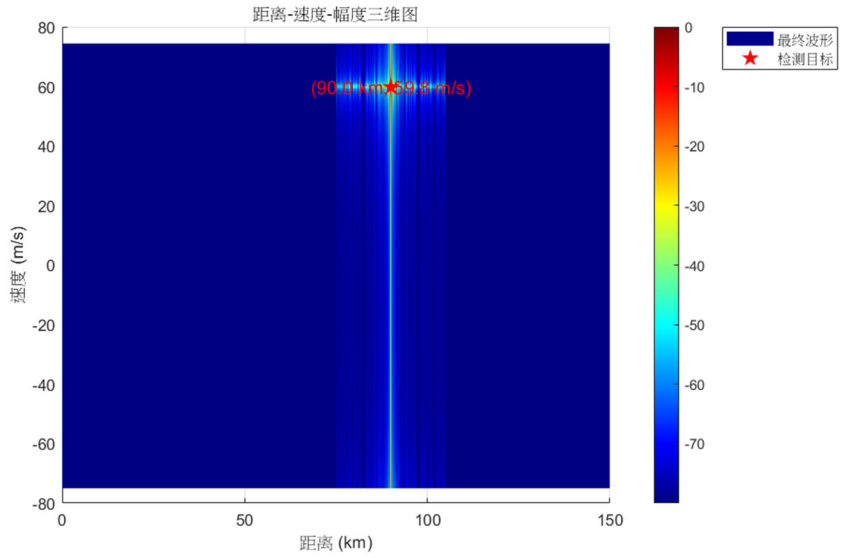


图 6.2 机载雷达接收信号处理后距离-速度-幅度谱的距离-速度平面图

调用编写的 CA-CFAR 函数进行恒虚警检测，输出结果如图 6.3、图 6.4 所示。此时累积脉冲数设置为 256，同前面仿真进行量化误差计算，可得其速度量化误差为±0.585m/s，相对误差范围为±0.975%，表明累积脉冲数增加可以提高检测性能，但代价是开销增加。

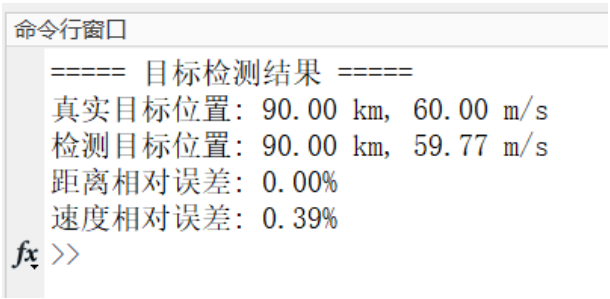


图 6.3 CA-CFAR 打印输出结果

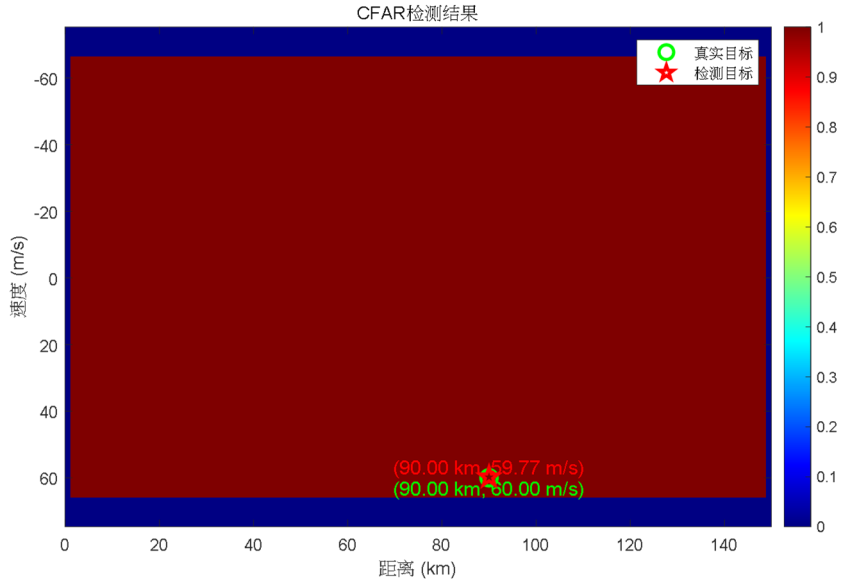


图 6.4 CA-CFAR 检测结果示意图

事实上，当接收机功率正好等于接收机背景噪声功率时，单阵元雷达方程计算的距离对应最大检测距离，即

$$R_{\max}=\left[\frac{P_t\lambda^2\sigma_t}{(4\pi)^3 L_s S_{i\min}}\right]^{0.25} \tag{2.16}$$

根据接收机输出检测所需最小信噪比，可以求出最小可检测信号信噪比

$$S_{i\min}=k_B T_0 BF\left(\frac{S}{N}\right)_{o\min} \tag{2.17}$$

根据信号噪声功率比表达式

$$\frac{S}{N}=\frac{S}{N_0 B}=\frac{ST_{au}}{N_0}=\frac{E_r}{N_0}\Rightarrow\left(\frac{S}{N}\right)_{o\min}=\left(\frac{E_r}{N_0}\right)_{o\min}=D_0 \tag{2.18}$$

可以得到最大检测距离

$$R_{\max}=\left[\frac{P_t T_{au}\lambda^2\sigma_t}{(4\pi)^3 k_B T_0 B D_0 F L_s}\right]^{0.25} \tag{2.19}$$

当雷达检测目标前进行多脉冲累积时，可改善信噪比，检波器输入端的 $D(N_{pulse})$ 值将下降，表明了雷达作用距离与累计脉冲数 N_{pulse} 的相关性。仿真实验也一定程度上体现了这一点。

改变目标速度（保持归一化多普勒频率下），绘制出 SCNR 曲线如图 7.1 所示。可以发现，SCNR 在目标速度小于 +70m/s 时保持极高的信杂比，当目标速度超过该范围时信杂比直线下降，表明机载雷达在 90km 处的检测目标最大速度应该为 +70m/s 。

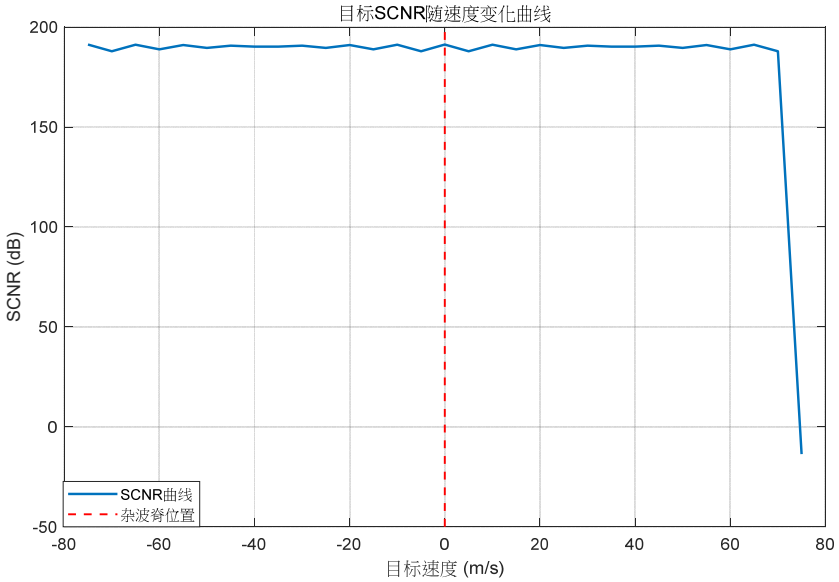


图 7.1 SCNR 关于目标速度的变化曲线

最后，该实验虽然还进行了 STAP 空时处理来代替 DBF-匹配滤波-MTD 处理过程，但仍然得到类似的结果。说明该仿真确实存在不太合理的地方，可能是建模方法等方面。受限于所学知识和学习精力，本人仅能勉强完成该仿真内容，对仿真结果进行一定地分析，但暂时无法确定问题所在处。

此外，鉴于先前基于深度学习的 IQ 信号处理的成功经历，日后将尝试引入一些深度学习方法来代替部分传统信号处理过程，在实现较好检测性能的同时减少繁琐地处理过程，避免一些方法的固有弊端，增强鲁棒性。

四、 附录 B

第二部分用到的仿真参数如下表所示。

参数	符号	数值	单位
光速	c	3×10^8	m/s
带宽	B	1	MHz
脉宽	T_{au}	100	μs
采样率	f_s	5	MHz
阵元数	N_R	16	-
单阵元功率	P_t	2×10^3	W
载机高度	H_c	5	km
载机速度	v_c	150	m/s
载波频率	f_c	1	GHz
波长	c	0.3	m
脉冲重复频率	PRF	1	kHz
累计脉冲数	N_{pluse}	256	-
阵元间距	d	0.15	m

接收机损耗系数	L_s	10	dB
噪声系数	F	5	dB
目标距离	R_t	90	km
目标速度	v_t	60	m/s
目标 RCS	σ_t	5	m^2
杂波后向散射系数	σ^0	0.01	-
杂波块个数	N_{bin}	101	-
标准温度	T_0	290	K
玻耳兹曼常数	k_B	1.38×10^{-23}	-
杂波距离环个数	k_{sam}	20	-
目标方位	φ_t	0	°
阵列与载机飞行方向夹角	φ_{array}	90	°
虚警概率	P_{fa}	1×10^{-6}	-
保护单元	$[G_r, G_d]$	$[5,5]$	-
参考单元	$[T_r, T_d]$	$[10,10]$	-

仿真实验中用到的主函数为 test2.m，自定义函数有：ClutterGen.m、cfar_2d.m 和 rx_array_airborneradar.m。

ClutterGen.m

```
%% ClutterGen.m 杂波建模函数
function [x_all, f_s, f_d, azimuth_c] = ClutterGen(H_c, R_t, v_c, azimuth_target, azimuth_array, ...
    N_bin, CPI_c, N_R, d, lambda, PRF, B, k_sam, sigma0, P_t, Ls)
    % 载机高度 H_c（m），目标距离 R_c（m），载机速度 v_c（m/s），
    % H_c 载机高度(m)
    % R_t 目标距离(m)
    % v_c 载机速度(m/s)
    % azimuth_target 目标方位（°）
    % azimuth_array 阵列与载机飞行方向夹角（°）
    % N_bin 杂波块个数
    % CPI_c CPI 内脉冲数
    % N_R 阵元数
    % d 阵元间隔（m）
    % lambda 波长(m)
    % PRF 脉冲重复频率(Hz)
    % B 带宽(Hz)
    % k_sam 样本个数（杂波距离环个数）
    % sigma0 杂波后向散射系数
```

```
    % P_t 发射功率 (W)
    % Ls 接收机损耗
    % 函数返回杂波信号 x_all （N_R×CPI，k_sam+1）

    % 常数设置
    c = 3e8;             % 光速 (m/s)
    H = H_c;       % 载机高度（m）
    v = v_c;             % 载机速度（m/s）
    L = CPI_c;           % CPI 内脉冲数
    delta_R = c / (2 * B); % 距离环间隔

    % % 添加噪声功率计算
    % k_B = 1.38e-23; % 玻尔兹曼常数
    % T0 = 290;        % 噪声温度
    % P_noise = k_B * T0 * B; % 噪声功率
    % CNR_dB = 40;
    % CNR_linear = 10^(CNR_dB/10);        % 1e4

    %--------------计算待测距离环和参考距离环的俯仰角----------------
    R = R_t;                         % 目标距离（m）
    R_all = R + delta_R * (-k_sam/2 : k_sam/2); % 所有距离环距载机的距离
    phi_all = asin(H ./ R_all);           % 所有距离环俯仰角

    % 杂波块数目和方向角设置
    azimuth_c = linspace(-pi/2, pi/2, N_bin); % 杂波块方位角/正侧视[-90,90]/正前视[0,180]
    theta_rel = azimuth_c - deg2rad(azimuth_array); % 考虑阵列方向
    d_theta = pi / (N_bin - 1);     % 方位角间隔

    %--------------各距离环杂波块的空时频率------------
    f_s = (d/lambda) * cos(phi_all)' * sin([linspace(-pi/2, 0, (N_bin-1)/2), 0, linspace(0, pi/2, (N_bin-1)/2)]); % 各距离环每个杂波块的空间频率
    f_d = (2*v/lambda) * cos(phi_all)' * sin([linspace(-pi/2, 0, (N_bin-1)/2), 0, linspace(0, pi/2, (N_bin-1)/2)]) / PRF; % 各距离环每个杂波块的多普勒频率

    Amplitude_all = zeros(k_sam+1,N_bin); % 各距离环各杂波块的复幅度
    x_all = zeros(N_R*L,k_sam+1);  % 所有距离环的杂波回波数据
    for ring_num = 1:length(R_all)
        R_ring = R_all(ring_num);
        phi = phi_all(ring_num);
        % 地面投影距离
        % R_ground = sqrt(R_ring^2 - H^2);
        R_ground = R_ring * cos(phi);

```

```
        %---------------计算各杂波块 CNR 和幅度---------------------
        area_patch = delta_R * R_ground * d_theta;
        RCS_patch = sigma0 * area_patch;

        % 雷达方程或者固定杂噪比计算幅度（天线方向图增益为 1）
        Pr = (P_t * N_R^2 * lambda^2 * RCS_patch) / ((4*pi)^3 * R_ring^4 * Ls)
    ; % CNR×P_noise 为杂波功率
        % Pr = CNR_linear * P_noise; % CNR×P_noise 为杂波功率

        Amplitude_all(ring_num,:) = sqrt(Pr); % 根据雷达方程或预设杂噪比计算该距
    离环每个杂波块的复幅度

        % 空时导向矢量
        a_s = exp(1j*2*pi*(0:N_R-1)' * f_s(ring_num,:));
        a_t = exp(1j*2*pi*(0:L-1)' * f_d(ring_num,:));

        %-----------------------回波数据-----------------------
        for clutterpatch_num = 1:N_bin
            x_all(:,ring_num) = x_all(:,ring_num) + ...
            Amplitude_all(ring_num,clutterpatch_num) * ...
            kron(a_t(:,clutterpatch_num),a_s(:,clutterpatch_num));
            %该距离环各杂波块回波数据叠加
        end
    end
end
```

cfar_2d.m

```
%% 二维 CA-CFAR 检测函数
function [detection_map, target_info] = cfar_2d(input_data, guard_win, train_w
    in, P_fa, range_bins, speed_bins, R_true, v_true)
    % input_data: 输入数据矩阵（距离门×速度门）
    % guard_win: 保护单元 [距离保护, 多普勒保护]
    % train_win: 参考单元 [距离参考, 多普勒参考]
    % P_fa: 虚警概率
    % range_bins: 距离轴向量
    % speed_bins: 速度轴向量
    % R_true: 真实目标距离
    % v_true: 真实目标速度

    [num_range, num_doppler] = size(input_data);
    detection_map = zeros(num_range, num_doppler);

    % 计算阈值因子
```

```
    num_ref_cells = (2*train_win(1)+2*guard_win(1)+1)*(2*train_win(2)+2*guard_win(2)+1) - ...
                    (2*guard_win(1)+1)*(2*guard_win(2)+1);
    alpha = num_ref_cells*(P_fa^(-1/num_ref_cells) - 1);

    % 滑动窗口检测
    for range_idx = 1+train_win(1)+guard_win(1) : num_range-train_win(1)-guard_win(1)
        for doppler_idx = 1+train_win(2)+guard_win(2) : num_doppler-train_win(2)-guard_win(2)
            % 定义检测区域
            range_win = range_idx-train_win(1)-guard_win(1):range_idx+train_win(1)+guard_win(1);
            doppler_win = doppler_idx-train_win(2)-guard_win(2):doppler_idx+train_win(2)+guard_win(2);

            % 提取参考单元
            ref_cells = input_data(range_win, doppler_win);

            % 去除保护单元
            ref_cells(train_win(1)+1:end-train_win(1), train_win(2)+1:end-train_win(2)) = NaN;

            % 计算噪声基底
            noise_level = mean(ref_cells(:), "omitmissing");
            threshold = alpha * noise_level;

            % 检测判决
            if input_data(range_idx, doppler_idx) > threshold
                detection_map(range_idx, doppler_idx) = 1;
            end
        end
    end

    % 匹配真实目标
    [~, true_range_idx] = min(abs(range_bins - R_true/1e3));
    [~, true_speed_idx] = min(abs(speed_bins - v_true));

    % 判断真实目标是否被检测到
    is_detected = false;
    if detection_map(true_range_idx, true_speed_idx) == 1
        detected_range = range_bins(true_range_idx);
        detected_speed = speed_bins(true_speed_idx);
        is_detected = true;
    end
```

```
55.
56.     if ~is_detected
57.         [detected_ranges, detected_speeds] = find(detection_map == 1);
58.         min_dist = inf;
59.         detected_range = NaN;
60.         detected_speed = NaN;
61.         for k = 1:length(detected_ranges)
62.             current_dist = sqrt(...
63.                 (range_bins(detected_ranges(k)) - R_true/1e3)^2 + ...
64.                 (speed_bins(detected_speeds(k)) - v_true)^2);
65.             if current_dist < min_dist
66.                 min_dist = current_dist;
67.                 detected_range = range_bins(detected_ranges(k));
68.                 detected_speed = speed_bins(detected_speeds(k));
69.             end
70.         end
71.     end
72.
73.     % ==== 错误处理：未检测到目标时给出警告 ====
74.     if isnan(detected_range)
75.         warning('未检测到真实目标，请调整 CFAR 参数!');
76.     end
77.
78.     % 计算误差
79.     if ~isnan(detected_range)
80.         range_error = abs(detected_range - R_true/1e3)/(R_true/1e3)*100;
81.         speed_error = abs(detected_speed - v_true)/abs(v_true)*100;
82.     else
83.         range_error = NaN;
84.         speed_error = NaN;
85.     end
86.
87.     % 输出结果
88.     target_info = struct(...
89.         'TrueRange', R_true/1e3, ...
90.         'TrueSpeed', v_true, ...
91.         'DetectedRange', detected_range, ...
92.         'DetectedSpeed', detected_speed, ...
93.         'RangeError', range_error, ...
94.         'SpeedError', speed_error);
95. end
```

rx_array_airborneradar.m

```
1.  %% 生成机载雷达均匀线阵阵列接收信号
```

```
function rx_array_signal = rx_array_airborneradar(St, array_phase, N_R, num_pulses, ...
    fd, PRF, td, N_PRT, N_d, N_st, P_t, RCS_t, R_t, lambda, Ls)
    % 参数设置
    % St LFM 发送信号
    % array_phase 阵列相位
    % N_R 阵元数
    % num_pulses 脉冲数
    % fd 多普勒频移
    % PRF 脉冲重复频率
    % td 目标延迟时间
    % N_PRT 单个 PRT 的采样点数
    % N_d 目标时延的采样点数
    % N_st 单个脉冲内的采样点数
    % P_t 单阵元功率(W)
    % RCS_t 目标 RCS(m²)
    % R_t 目标距离(m)
    % lambda 波长(m)
    % Ls 接收机损耗
    % 返回阵列接收信号 （脉冲数×单个 PRT 的采样点数，阵元数）

    % 计算目标接收功率（单阵元雷达方程）
    Pr = (P_t * lambda^2 * RCS_t) / ((4*pi)^3 * R_t^4 * Ls);
    A_t = sqrt(Pr);  % 目标信号幅度

    % % 生成幅度加权的发射信号
    % St_scaled = A_t * St;

    % 初始化多通道接收信号
    rx_array_signal = zeros(num_pulses * N_PRT, N_R) * A_t;
    for n = 0:num_pulses-1
        % 生成单脉冲回波（含多普勒相位和时间延迟）
        doppler_phase = exp(1j * 2 * pi * fd * (n / PRF + td));
        % 生成阵列接收信号
        for k = 1:N_R
            % 每个阵元的相位补偿
            R_phase = array_phase(k) * doppler_phase;
            % 计算信号位置
            start_idx = n * N_PRT + N_d;
            end_idx = start_idx + N_st;
            % 截断处理
            if end_idx > num_pulses * N_PRT
                end_idx = num_pulses * N_PRT;
                valid_len = end_idx - start_idx;
                rx_array_signal(start_idx+1:start_idx+valid_len, k) = ...
```

```
                    rx_array_signal(start_idx+1:start_idx+valid_len, k) + ...
                    St(1:valid_len).' * R_phase;
            else
                rx_array_signal(start_idx+1:end_idx, k) = ...
                    rx_array_signal(start_idx+1:end_idx, k) + ...
                    St.' * R_phase;
            end
        end
    end
end
```